# 在数据库中挖掘定量关联规则的方法研究[1]

程 岩，卢 涛，黄梯云

（哈尔滨工业大学管理学院，哈尔滨 150001）

**摘要**：数据挖掘技术是实现智能决策支持系统的一个重要手段，关联规则是数据挖掘的一个重要内容，传统的 Apriori 算法仅适用于挖掘数据间的定性关联关系，但数据间的定量关联关系对决策更有帮助，属性值的离散映射是挖掘定量关联规则的一个重要环节，离散映射中属性值区间的划分粒度是影响数据挖掘质量的一个重要因素，本文结合粗集理论提出了一个确定属性值划分粒度的方法，在此基础上设计出一个挖掘定量关联规则的算法：Apriori _ 2，利用 Apriori _ 2 可以挖掘出大量对决策有帮助的定量关联规则。

**关键词**：数据挖掘；智能决策支持系统；关联规则；粗集

**中图分类号**：TP311　　**文献标识码**：A　　**文章编号**：1007-9807(2001)04-0041-08

## 0 引 言

随着信息系统的建设和发展，许多企业和组织积累了大量的数据，而数据本身不是信息，隐含在数据中的规则、模式等知识才是对决策有帮助的信息。数据挖掘的目的就是发现隐含在数据中对决策有帮助的信息，它是实现智能决策支持系统的一个重要手段[1]。

关联规则是数据挖掘的一个重要内容，通常关联规则反映的是数据间的定性关联关系。表 1 为一个商品交易数据库，一条记录表示用户一次购买的商品种类，每个属性（A，B…）代表一种商品，每个属性都是布尔类型的。一条关联规则的例子是：{A，B}→{D}[2%][60%]，规则的含义是"如果用户购买商品 A 和 B，那么也可能购买商品 D，因为同时购买 A，B 和 D 的交易记录数占总交易数的 2%，而购买 A 和 B 的交易中，有 60% 的交易也包含 D"。规则中 60% 是规则的信任度，2% 是规则的支持度[2—6]。数据挖掘就是要发现所有满足用户定义的最小信任度和支持度阈值限制的关联规则。表 1 中的数据只是定性地描述一个交易是否包含某商品，而对交易量没有定量描述，这种布尔类型数据间的关联规则被称为定性关联规则。但数据记录的属性往往是数值型或字符型的，这些数据间也存在对决策有帮助的关联规则，相对于定性关联规则，这些规则被称为定量关联规则[3—7]。

表 1　商品交易表

| ID | A | B | C | D | E | … |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 2 | 1 | 1 | 1 | 1 | 0 | 0 |
| … | … | … | … | … | … | … |

在定量关联规则的挖掘中主要有两个步骤：

（1）将属性值按一定的划分区间映射为标准的离散值，这个过程的一个主要问题是如何确定合理的属性值划分区间。

（2）从离散映射后的数据集中挖掘出定量关联规则。由于属性值的划分区间不只有两个，属性值不能仅仅离散映射为 0，1 两个值，因此不能用传统的定性关联规则的挖掘算法挖掘定量关联规则。

针对上述两个步骤中存在的问题，结合粗集

[1] 收稿日期：1999-11-22；修订日期：2000-04-18。
基金项目：国家自然科学基金资助项目（70071006）
作者简介：程 岩（1967-），男，博士生。

理论和概念树理论提出了一个确定属性值划分区间的算法，并在挖掘定性关联规则的 Apriori 算法基础上设计出一个挖掘定量关联规则的算法 Apriori_2.

# 1 基于粗集理论挖掘定量关联规则的基本原理

## 1.1 关系数据库的粗集模型

令 $DB=(U,C,D,V,f)$ 表示一个关系数据库，其中 $U=\{u_1,u_2,\cdots,u_N\}$ 是数据记录的集合；数据记录的属性间往往存在决定和被决定的关系，令 $C$ 为决定属性的集合，称为条件属性集，$D$ 为被决定属性的集合，称为决策属性集，且要求 $C\cap D=\varnothing$；令 $A=C\cup D$；$V$ 表示所有属性可取值的集合；$f:U\times A\to V$ 是一个信息函数，该函数为某个数据记录的某个属性赋予一个特定的值，例如 $f(u_i,a)=\lambda$，表示记录 $u_i$ 在属性 $a$ 上的取值为 $\lambda$.

数据记录间往往存在不分明关系，令 $B\subseteq A$，属性集 $B$ 上的不分明关系 $R(B)$ 定义为 $R(B)=\{(u_i,u_j)\in U^2|\forall a\in B, f(u_i,a)=f(u_j,a)\}$. 不分明关系是集合 $U$ 上的等价关系，它将集合 $U$ 划分为一些等价类. 令 $u_i\in U$，记录 $u_i$ 在属性集 $B$ 上的等价类定义为 $R_B(u_i)=\{u_i\in U|\forall a\in B, f(u_i,a)=f(u_j,a)\}$. 集合 $U$ 上的等价类又称为基本集，基本集 $R_B(u_i)$ 中任一个数据记录在属性集 $B$ 上所取的所有属性值称为该基本集的描述[5,9]，区分两种基本集：特征集和概念.

**定义 1** 如果确定基本集 $R_B(u_i)$ 的属性集 $B$ 满足：$B\subseteq C$，即基本集是条件属性集上的等价类，称这一类基本集为特征集，记为 $[X]$，其中 $X$ 是特征集 $[X]$ 的描述；如果 $B\subseteq D$，即基本集是决策属性集上的等价类，称该基本集为概念，记为 $[Y]$，其中 $Y$ 是概念 $[Y]$ 的描述.

**定义 2** 令 $[X]$ 是不分明关系 $R(B)$ 确定的一个特征集，$B\subseteq C$，如果属性集 $B$ 中有 $k$ 个属性，那么称 $[X]$ 为 $k$ 元特征集，记为 $[X^k]$，其中 $X^k$ 称为 $k$ 元特征描述.

**定义 3** 令 $X^k=(C_1=\lambda_1)\wedge\cdots\wedge(C_k=\lambda_k)$ 是一个 $k$ 元特征描述，其中 $C_i\in C,\lambda_i\in V$，$k$ 元特征描述 $X^k$ 的表达式中的任一个等式 $(C_i=\lambda_i)$ 称为 $X^k$ 的一元子句；令 $1\le i<j\le k$，$X^k$ 中的表达式 $(C_i=\lambda_i)\wedge\cdots\wedge(C_j=\lambda_j)$ 称为描述 $X^k$ 的子句.

## 1.2 基于粗集模型挖掘定量关联规则

决策者关心的仅仅是条件属性值与决策属性值间的关联关系，因此一个关联规则的前提对应一个特征集的描述 $X$，一个规则的结论对应一个概念的描述 $Y$，定量关联规则表示为 $X\to Y[s][c]$，规则中 $s,c$ 分别代表规则的支持度和信任度. 通过计算特征集 $[X]$ 的支持度和信任度就可以判断关联规则 $X\to Y$ 是否成立. 根据关联规则的定义，特征集 $[X]$ 的支持度与信任度的计算公式为

$$s=\frac{\mathrm{card}([X]\cap[Y])}{\mathrm{card}([U])},\quad c=\frac{\mathrm{card}([X]\cap[Y])}{\mathrm{card}([X])}$$

式中运算符 card() 是求集合中元素的个数. 令 $\alpha$ 为用户定义的最小支持度阈值，$\beta$ 为最小信任度阈值. 根据关联规则的定义：当 $s>\alpha$ 且 $c>\beta$ 时，规则 $X\to Y$ 成立.

## 1.3 属性值的离散映射

在挖掘关联规则前，首先要将属性的取值映射为标准的离散符号，规定每个离散符号表示一个数据区间. 例如在城市供水信息系统的数据库中有属性 maximum_temperature 和 date，可以按区间 $[-40℃,-35℃]$，$[-35℃,-30℃]$… 将 maximum_temperature 的值分别映射为 0,1…. 字符型属性值也可以按一定的区间映射为离散符号，例如按较大的区间可以将属性 date 的取值按 work_day，rest_day 分别映射为 0,1，也可以按较小的区间将 date 的取值按 Monday，Tuesday… 进行映射. 称离散映射时属性值区间划分的大小为属性值的划分粒度. 通过属性值的离散映射，数据库中的每一条记录都由标准的离散值表示.

在属性值的离散映射过程中，条件属性值的划分区间过小会使特征集包含的元素变少，许多规则因支持度低于阈值而不成立，这类问题称为最小支持度问题；而条件属性值区间的划分粒度过大会使许多规则因信任度过小而不能成立，这类问题称为最小信任度问题.

## 2　条件属性值划分区间的确定

### 2.1　属性概念树

对某个属性依其所有取值在层次上进行分类,直到不能再分为止,这样形成的概念层次结构称为该属性的概念树.图1是属性date的概念树.

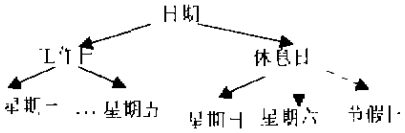


图1　属性日期的概念树

属性概念树的形成有三种方式:自动生成、半自动生成和人工定义.在数值型属性的概念树上,每一个节点表示一个数值区间,可以根据数据的分布特征采用统计方法自动生成数值型属性的概念树[10].例如,某城市在某季节的最高气温大部分分布在20℃—30℃间,且历史上的最大值35℃,最小值为10℃,这时可以将数据划分为三个区间:[10℃,20℃],[20℃,30℃],[30℃,35℃].再按同样的方法对每一个区间进行详细的划分,直到形成一个完整的概念树.某些字符型属性可以利用数据库的结构自动形成概念树,例如数据库的结构为:(country,province,city,…),这种结构便体现了属性city的概念层次,通过计算可以很容易形成city的概念树;如果经过统计分析发现属性country的值大多数为"china",这时可以人为定义一个概念层次"foreign",然后再通过算法自动形成city的概念树,这种方式便是半自动生成方式[11].有些属性只能由专家根据应用环境的要求进行人工定义.

属性概念树在数据挖掘过程中具有巨大的应用价值,但使用者需要很大的努力才能够生成属性概念树,不过这种工作只需做一次便可以满足以后所有数据挖掘的需要,这是因为属性概念树作为一种背景知识具有稳定性,因此用户的这种努力是值得的[11].

### 2.2　确定属性值划分区间的算法

在属性概念树上,子辈节点代表的数值区间是对其父辈节点代表的数值区间的详细划分,节点层次越高表示属性值的划分区间越大.属性值离散映射过程中的区间划分问题就是为每个属性在其概念树上选取一个合适的节点层次进行属性值的离散映射的问题.

首先将每个属性的离散值取它的概念树的最底层节点,然后对属性的离散值进行适当的概念爬升,直到为每个属性找到一个合适的节点层次.所谓属性概念爬升就是在数据库中用属性概念树上高层节点代替该属性的低层节点进行属性值的离散映射.但随着概念爬升,属性值区间的划分粒度越来越大,数据挖掘过程中出现最小信任度问题的可能性也越来越大,下面通过粗集理论定量分析出现最小信任度问题的可能性大小.

设有数据库$DB=(U,C,D,V,f)$,且$\mathrm{card}(C)=m$,$[Y]$表示$DB$中的一个概念,所有其它概念记为$[\neg Y]$,即$[\neg Y]=U-[Y]$,$[X^m]$表示由不分明关系$R(C)$确定的一个$m$元特征集.令$\beta$为最小信任度阀值,$c$表示规则$X^m\to Y$的信任度,$c'$表示规则$X^m\to\neg Y$的信任度,根据信任度的计算公式有

$$\begin{aligned}c'&=\frac{\mathrm{card}([X^m]\cap[\neg Y])}{\mathrm{card}([X^m])}\\&=\frac{\mathrm{card}([X^m]\cap(U-[Y]))}{\mathrm{card}([X^m])}\\&=\frac{\mathrm{card}([X^m]-[X^m]\cap[Y])}{\mathrm{card}([X^m])}\\&=1-\frac{\mathrm{card}([X^m]\cap[Y])}{\mathrm{card}([X^m])}\\&=1-c\end{aligned}$$

如果$X^m\to\neg Y$的信任度$c'>\beta$,必有$(1-c)>\beta$,即$c<(1-\beta)$.根据规则$X^m\to Y$的信任度$c$的取值,将数据库中的所有$m$元特征集划分为三类:(1)$c>\beta$的$m$元特征集.(2)$c<(1-\beta)$(即$c'>\beta$)的$m$元特征集.(3)$\beta\geq c\geq(1-\beta)$的$m$元特征集(注:在实际应用中$0.5<\beta<1$,所以$\beta>(1-\beta)$).

根据这三类特征集,将记录集$U$划分为三个互不相交的区间,即概念$[Y]$的$\beta$正例区域,概念$[Y]$的$\beta$边界区域和概念$[Y]$的$\beta$反例区域:概念$[Y]$的$\beta$正例区域是所有第1类$m$元特征集的并集,记为$POS_\beta(Y)=\bigcup\{[X^m]\subseteq U\mid c>\beta\}$;概念$[Y]$的$\beta$反例区域是所有第2类$m$元特征集并集,记为$NEG_\beta(Y)=\bigcup\{[X^m]\subseteq U\mid c<(1-\beta)\}$;概念$[Y]$的$\beta$边界区域是所有第3类$m$元特征集并集,记为$BIND_\beta(Y)=\bigcup\{[X^m]\subseteq U\mid\beta\geq c\geq(1-\beta)\}$.

因为 $BIND_\beta(Y)$ 中的特征集的信任度都小于阈值，所以如果概念 $Y$ 的 $POS_\beta(Y)$ 很小而 $BIND_\beta(Y)$ 很大，许多结论为概念描述 $Y$ 的规则会因信任度小于阈值而不能成立. 可见 $POS_\beta(Y)$ 与 $BIND_\beta(Y)$ 的大小是决定数据挖掘时是否出现最小信任度问题的一个重要因素. 用一个数值指标 $\gamma(Y)$ 来定量地描述在一个数据集中挖掘结论为 $Y$ 的定量关联规则时出现最小信任度问题的可能性的大小，$\gamma(Y)$ 的计算公式为

$$\gamma(Y)=1-\frac{\mathrm{card}(BIND_\beta(Y))}{\mathrm{card}(POS_\beta(Y)\cup BIND\beta(Y))}.$$

显然，$\gamma(Y)$ 越大越不可能出现最小信任度问题，反之就越可能出现最小信任度问题.

随着概念爬升的进行，$\gamma(Y)$ 会越来越小，从而会提高出现最小信任度问题的可能性，用 $\gamma(Y)$ 的变化率 $\Delta\gamma$ 来定量地描述这种可能性的变化情况，$\Delta\gamma$ 的计算公式为 $\Delta\gamma=\frac{(\gamma'(Y)-\gamma(Y))}{\gamma(Y)}$. 公式中 $\gamma(Y)$ 为最初用属性概念树最底层节点离散映射后的数据集的指标 $\gamma(Y)$ 的值，$\gamma'(Y)$ 为每次属性概念爬升后的指标 $\gamma(Y)$ 的值. 可以根据 $\Delta\gamma$ 的数值判断属性值是否已经爬升到一个合适的概念层次. 假设有一个经验数值 $\eta$，$\eta$ 表示属性概念爬升到合适概念层次时 $\gamma(Y)$ 的变化率（$\eta$ 是接近 0 的小数，如何确定 $\eta$ 见后文的实例分析），判断是否停止概念爬升的标准就是判断 $\Delta\gamma$ 是否大于 $\eta$.

为确保每个属性都能找到一个合适的概念层次，属性的概念爬升遵守两个原则：(1) 每次属性值都只提升一个概念层次. (2) 每次只选择一个使 $\gamma(Y)$ 变化最小的属性进行概念爬升.

根据以上原理设计出通过概念爬升确定条件属性值区间划分粒度的算法：Value _ partition. 由于算法是针对每一个概念描述 $Y$ 依次进行挖掘的，因此针对每一个概念描述 $Y$ 都用算法 Value _ partition 输出一个新的属性值经过合适的离散映射的数据库，再对输出的数据库进行计算，挖掘出结论为概念描述 $Y$ 的所有关联规则.

Algorithm：Value _ partition

输入：数据库 $DB$，条件属性的概念树 $H$，一个概念描述 $Y$，允许的 $\gamma(Y)$ 的变化率阈值 $\eta$，关联规则信任度阈值 $\beta$，条件属性集 $C$

输出：一个新的属性值经过离散映射的数据库 $DB'$

begin

用条件属性概念树最底层节点将 $DB$ 离散映射为 $DB'$

计算 $DB'$ 的 $\gamma(Y)$ 的值，记为 $\gamma'(Y)$

$\Delta\gamma\leftarrow 0$ //令 $\gamma(Y)$ 的变化率指标 $\Delta\gamma$ 的初始值为 0.

$S_1\leftarrow\Phi$，$S_2\leftarrow\Phi$ //$S_1$，$S_2$ 为两个临时表，$\Phi$ 为空集;

While $\Delta\gamma\leqslant\eta$ do

【$\gamma'(Y)\leftarrow 0$ //$\gamma'(Y)$ 为概念爬升后指标 $\gamma(Y)$ 的值，首先将 $\gamma'(Y)$ 初始化为 0;

for $C$ 中的每个属性 $C_i$ do //通过该循环找出一个使 $\gamma(Y)$ 的值变化最小的属性进行概念爬升;

【$S_2\leftarrow DB'$ //$S_2$ 是 $DB'$ 的一个临时备份，以便进行计算. 每循环一次，都将 $DB'$ 赋给 $S_2$;

if 属性 $C_i$ 的概念树上有更高层数据可用于替换当前数据库中的数据 then

【用 $C_i$ 的概念树上高一层次数据替换数据库 $S_2$ 中的数据计算 $S_2$ 的指标 $\gamma(Y)$ 的值，记为 $\gamma_2(Y)$

if $\gamma_2(Y)>\gamma'(Y)$ then

【$S_1\leftarrow S_2$ //用临时表 $S_1$ 记载使 $\gamma(Y)$ 变化最小的属性概念爬升后的数据集;$\gamma'(Y)\leftarrow\gamma_2(Y)$】

】

】

$DB'\leftarrow S_1$ //用 $S_1$ 替换 $DB'$，表示 $DB'$ 进行了一次概念爬升;

$\Delta\gamma\leftarrow(\gamma'(Y)-\gamma(Y))\div\gamma(Y)$ //计算属性概念爬升后 $DB'$ 的指标 $\gamma(Y)$ 的变化率;

】

End (Value _ partition)

算法时间复杂度：设数据库有 $N$ 条记录，$m$ 个条件属性，每个属性的概念树有 $H$ 个层次. 对每一个属性，用高层概念数据替换低层概念数据的计算时间为 $N$，指标 $\gamma(Y)$ 的计算时间为 $N\log N$，对一个属性进行概念爬升并计算 $\gamma(Y)$ 的计算时间为 $N+N\log N$. 因为数据库每进行一次概念爬

升前都要分别对 $m$ 个属性进行概念爬升，选出一个最优的属性进行概念爬升，因此数据库每进行一次概念爬升的计算时间为 $m \cdot (N + N\log N)$。数据库最多可进行 $H \cdot m$ 次概念爬升，因此总的计算时间为 $m^2 \cdot H \cdot (N - N\log N)$。因为数据库中 $H$，$m$ 远远小于 $N$，因此算法的时间复杂度为 $O(N\log N)$。

## 3　关联规则的发现算法

属性值经过离散映射后，就可以通过计算每个特征集的支持度和信任度发现结论为概念描述 $Y$ 的所有关联规则，但这种方法有两个缺点：1）计算效率低：这是因为特征集的数量非常大，假设有 $m$ 个条件属性，每个属性可取 $L$ 个离散值，那么一元特征集的数量为 $C_m^1 \cdot L$，$k$ 元特征集的数量为 $C_m^k \cdot L^k$，特征集的总数为 $C_m^1 \times L + C_m^2 \times L^2 + \cdots + C_m^m \times L^m$。2）计算出的规则解释能力低，数据挖掘的结果包含大量冗余规则。

**定义4**　设 $X_1 \rightarrow Y$，$X_2 \rightarrow Y$ 是两条结论都为概念描述 $Y$ 的规则，如果 $X_2$ 是 $X_1$ 的子句，那么 $X_2 \rightarrow Y$ 是一条冗余规则。

例如有如下两条规则：(20℃ < maximum_temperature ≤ 30℃) → (600 000 < water_demand ≤ 650 000 ton)(20℃ < maximum temperature ≤ 30℃) ∧ (60% < average_humidity ≤ 75%)) → (600 000 < water_demand ≤ 650 000 ton) 如果第一条规则成立，第二条规则必然成立，因此第二条规则就是冗余的。

为了提高计算效率，在 Apriori 算法基础上设计了一个挖掘定量关联规则的算法：Apriori_2。Apriori_2 的主要原理是：当计算结论为 $Y$ 的规则时，不需要计算所有特征描述与概念描述间的支持度与信任度，而是首先从1元特征集开始计算，如果判断出某个特征集描述 $X$ 与 $Y$ 间的支持度小于阀值 $\partial$，那么所有以 $X$ 为子句的特征描述与 $Y$ 间的支持度必然小于 $\partial$，这样就不必对这些特征集进行计算了。

Apriori_2 通过多步递推运算排除大量不必进行计算的特征集，每一步递推运算包括两个阶段：(1) 生成候选集：为了减少计算量，在第 $k$ 步递推运算中利用第 $k-1$ 步计算的结果，将不能使规则成立的 $k$ 元特征集判断出来，将剩下的 $k$ 元特征集作为候选特征集存入集合 $R_k$ 中。(2) 从 $R_k$ 中计算出所有支持度大于 $\partial$ 的 $k$ 元特征集，并将这些 $k$ 元特征集存入 $L_k$ 中，再计算 $L_k$ 中的每个特征集与 $Y$ 间的信任度，将信任度大于阀值的规则作为知识存入规则库中。反复进行递推计算，当 $L_k$ 为空时算法停止计算。

**定义5**　令 $\partial$ 是关联规则最小支持度阀值，称支持度大于 $\partial$ 的特征集为大特征集。令 $L_k$ 是 $k$ 元大特征集的集合，$R_k$ 是 $k$ 元特征集的集合，且 $R_k$ 是 $L_k$ 的超集，称 $R_k$ 为 $k$ 元候选集。

**定理1**　特征集是大特征集的必要条件是其描述的所有子句确定的特征集是大特征集。

**证明**　设 $X^{k-1}$ 是 $X^k$ 的任一子句，根据特征集的定义有 $[X^k] \subseteq [X^{k-1}]$，因此如果 $[X^k]$ 的支持度大于阀值 $\partial$，$[X^{k-1}]$ 的支持度必然大于 $\partial$；反之，如果 $[X^{k-1}]$ 的支持度不大于 $\partial$，$[X^k]$ 的支持度必然不大于 $\partial$。　　证毕

1元候选集 $R_1$ 是所有一元特征集的集合，算法通过 $R_1$ 可以计算出 $L_1$，再根据 $L_1$ 计算出 $R_2$，如此递推，要解决的一个问题是如何根据 $L_{k-1}$ 构造候选集 $R_k$，使 $R_k$ 满足：1）$R_k$ 的规模尽量小，2）$R_k$ 中包含所有使规则成立的 $k$ 元特征集，而不包含使冗余规则成立的特征集。

通过对 $L_{k-1}$ 中的特征集进行交运算生成候选集 $R_k$，再对 $R_k$ 进行删除运算使 $R_k$ 的规模尽量小，同时又不包含使冗余规则成立的特征集。

1）特征集的交运算

设 $[X_1]$、$[X_2]$ 是两个特征集，两个特征集进行交运算后生成的特征集的描述 $X$ 是 $X_1$、$X_2$ 中所有子句的合取范式，例如 $[(C_1 = 1)] \wedge (C_2 = 0)] \cap [(C_3 = 2)] = [(C_1 = 1) \wedge (C_2 = 0) \wedge (C_3 = 2)]$。

根据定理1，如果一个特征集不是大特征集，那么该特征集与任何特征集进行交运算所生成的特征集都不是大特征集，因此只需对 $L_{k-1}$ 中的特征集进行交运算生成候选集 $R_k$。这里规定：在描述的公式中，一元子句间的排列顺序按属性的字母顺序排列，例如 $(C_1 = 1) \wedge (C_2 = 0)$ 不能写为 $(C_2 = 0) \wedge (C_1 = 1)$。为了使生成的候选集中不会漏掉使规则成立的特征集，又要保证对 $L_{k-1}$ 中

的特征集进行交运算生成的特征集都是 $k$ 元的，规定两个特征集进行交运算的条件是：①两个特征集的描述中前 $(k-2)$ 个一元子句相同，②第 $(k-1)$ 个一元子句中的描述属性是两个不同的属性。例如有 $L_2$ 如下：

$\{[(C_1=1)\wedge(C_2=0)],[(C_1=1)\wedge(C_3=1)],[(C_1=1)\wedge(C_3=2)],[(C_2=1)\wedge(C_3=2)],[(C_2=1)\wedge(C_3=3)]\}$

在 $L_2$ 中，第 1 个特征集可以和第 3 个特征集进行交运算形成一个 3 元特征集：$[(C_1=1)\wedge(C_2=0)\wedge(C_3=2)]$，由于不符合交运算的第 1 个条件，第 2 个特征集不能与第 5 个特征集进行交运算，尽管交运算的结果是 3 元特征集：$[(C_2=1)\wedge(C_3=1)\wedge(C_3=3)]$，这是因为：根据定理 1 的原理：如果 $[(C_1=1)\wedge(C_2=1)\wedge(C_3=3)]$ 是大特征集，它的 2 元子句确定的特征集 $[(C_2=1)\wedge(C_3=3)]$ 必然在 $L_2$ 中，否则这个 3 元特征集就不是大特征集。第 1 个特征集也不可以和第 2 个特征集进行交运算，这是因为两个特征集的描述中的第 2 个子句中的属性都是 $C_3$，不符合交运算的第 2 个条件，这两个特征集进行交运算的结果不是 3 元特征集。$L_2$ 经过交运算后，生成一个 3 元候选集 $R_3$ 为 $\{[(C_1=1)\wedge(C_2=0)\wedge(C_3=2)],[(C_1=1)\wedge(C_2=1)\wedge(C_3=2)]\}$

2）删除运算

根据定理 1 的原理，算法对 $R_k$ 中的每个特征集 $[X^k]$ 的所有 $(k-1)$ 元子句进行检查，如果发现有某个 $(k-1)$ 元子句所确定的特征集 $[X^{k-1}]$ 不在 $L_{k-1}$ 中，便将该特征集从 $R_k$ 中删除。

计算出候选集 $R_k$ 后只需计算 $R_k$ 中每一个特征集的支持度便可以计算出 $R_k$ 中的大特征集，并形成 $L_k$。计算 $L_k$ 中每一个大特征集的信任度，如果信任度大于阈值 $\beta$，便挖掘出一个关联规则：$X^k\rightarrow Y$。挖掘出一个规则 $X^k\rightarrow Y$ 后，为避免产生冗余规则，将该特征集 $[X^k]$ 从 $L_k$ 中删除。根据生成候选集的交运算和删除运算的原理，将特征集 $[X^k]$ 从 $L_k$ 中删除后就可以确保 $R_{k+1}$ 中的特征集的描述不包含子句 $X^k$，这样就避免了冗余规则的出现。

根据以上原理，设计出一个定量关联规则的发现算法 Apriori_2，如下所示：

Algorithm : Apriori_2

输入：数据库 $DB$，关联规则的最小支持度阈值 $\alpha$，信任度阈值 $\beta$，所有条件属性的概念树 $H$，允许的 $T(Y)$ 的变化率阈值 $\mu$，条件属性集 $C$，决策属性集 $D$

输出：所有定量关联规则

```
begin
  For  每一个概念描述 Y  do
  【value_partition //生成一个属性值为标准离散值的数据库 DB' 用来计算概念[Y] 的关联规则;rule_mine //挖掘规则,rule_mine 见下面的过程;将规则库中用离散符号表示的规则转变为用实际属性值表示的规则】
  End {Apriori_2}
Procedure:rule_mine
  begin
    根据 DB' 的粗集模型计算出所有一元特征集,形成一元候选集 R1
    k←1
    judge←true  {用变量 judge 来标识 Lk 是否为空,当 judge = false 时表示 Lk 是空集}
    While judge ≠ false do {即 Lk 不是空集时运行以下步骤}
    【  计算 Rk 中每一个特征集的支持度,将支持度大于 α 的特征集存入集合 Lk 中
      for  Lk 中每个 k 元大特征集[X^k] do
        【  计算[X^k] 的信任度
          if  [X^k] 的信任度大于阈值 β,then
            【  将[X^k] 从 Lk 中删除
                将 X^k → Y 存入规则库中】
        】
      if  Lk ≠ Ø then  {Ø 表示空集}
          【对 Lk 中的特征集进行交运算生成 Rk+1;
            对 Rk+1 中的特征集进行删除运算】
      else judge←false
      k←k + 1
    】
End {rule_mine}
```

算法时间复杂度：设数据库中共有 $N$ 条记录，$m$ 个条件属性，$h$ 个概念，每个属性可取 $L$ 个离

散值；计算出一个特征集的支持度和信任度的时间复杂度为 $O(N)$，数据集共有 $K=C_m^1\cdot L+C_m^2\cdot L^2+\cdots+C_m^m\cdot L^m$ 个特征集，在最坏情况下计算某个概念 $Y$ 的关联规则需要计算 $K$ 次支持度和复杂度，而属性值离散映射的时间复杂度为 $N\log N$，因此算法的计算时间为 $h\cdot(K\cdot N+N\log N)$，因为数据库中 $m,L,h$ 远远小于 $N$，因此算法的时间复杂度为 $O(N\log N)$.

## 4　实例分析

为了降低供水成本，城市生活供水的调度系统往往根据专家的经验判断每天生活用水量，但仅仅根据主观经验不能保证决策的质量. 根据 Apriori_2 的原理，开发了一个挖掘定量关联规则的挖掘工具，并将其应用到城市生活供水的决策支持系统中.

根据专家经验，每天城市生活用水量需要根据如下 7 个条件属性判断：最高气温、最低气温、平均湿度、平均风速、降雨量、降雪量及日期. 根据供水公司实际工作需要，将需水量分为 10 个级别，分别用 0,1,…,9 来表示. 日期的概念树由专家定义，如图 1 所示. 其它条件属性的概念树通过统计的方法自动获得. 在算法 Apriori_2 中关联规则支持度阈值 $\alpha$，信任度阈值 $\beta$ 和允许的 $\Upsilon(Y)$ 的变化率阈值 $\mu$ 是用户主观输入的经验参数. 为了确定合理的参数值以及测量数据挖掘结果的质量，将沈阳市过去几年的天气及用水量数据划分为两部分，80% 的数据用于挖掘关联规则，20% 的数据用于检验数据挖掘结果的质量. 用不同的参数进行了 8 次运算，获得了不同的预测结果，计算结果如表 2 所示. 显然，第 5 次运算所采用的参数可以获得比较理想的预测结果.

表 2　实验结果

| 参数值 | | | 精确度 | 规则数 |
|---|---|---|---|---|
| $\alpha$ | $\beta$ | $\mu$ | | |
| 3% | 90% | 0 | 61.1% | 131 |
| 3% | 75% | 0.05 | 65.3% | 201 |
| 4% | 75% | 0.05 | 74.1% | 186 |
| 4% | 80% | 0.03 | 81.2% | 173 |
| 6% | 80% | 0.03 | 82.3% | 142 |
| 6% | 85% | 0.01 | 81.7% | 134 |
| 8% | 85% | 0.02 | 79.7% | 129 |
| 7% | 85% | 0.01 | 80.5% | 135 |

## 5　结论

知识的自动获取是智能决策支持系统的一个重要环节，Apriori_2 算法的一个重要优势是：它通过发现属性值间的定量关联关系从粗糙(Rough)数据中获得分类和预测知识. 传统的示例学习算法也是获取分类知识的一个重要手段，但这些算法只能从精确数据集中获取知识，这样就限制了它们的应用范围. 信息的不完备性是管理领域中半结构化决策问题和非结构化决策问题的共同特点，由于信息的不完备所导致的数据集的粗糙性是普遍存在的现象，因此 Apriori_2 算法在管理决策领域具有更广泛的应用前景.

**参考文献：**


[1] Chen M S, Han J W, Yu P S. Data mining: an overview from a database perspective[J]. IEEE Transactions on Knowledge and Data Engineering, 1996, 8(6): 866-883

[2] Piatetsky-Shapiro G. Discovery, analysis and presentation of strong rules[C]. In Piatetsky-Shapiro G, Frawley W J ed. Knowledge Discovery in Databases, AAAI/MIT Press, Cambridge, 1991. 229-248

[3] Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large databases[C]. In Buneman P, Jajodia S ed. Proceeding of the ACM SIGMOD Conference on Management of Data. ACM Press, Washington, D.C., 1993. 207-216

[4] Brin S, Motwani R, Ullman J D. Dynamic itemset counting and implication rules for market basket data[C]. In Peckham J ed. Proceeding of the ACM SIGMOD Conference on Management of Data. ACM Press, Tucson, Arizona, USA, 1997. 255-264

[5] Houtsma M, Swami A. Set-oriented mining for association rules in relational databases[C]. In Yu P S, Chen A L P

ed. IEEE 11th International Conference on Data Engineering. IEEE Computer Society Press. Los Alamitos, California. 1995. 25-33

[6] Srikant R. Vu Q. Agrawal R. Mining association rules with item constraints[C]. In Simoudis E. Han J W. Fayyad U ed. Proceedings of Second International Conference on Knowledge Discovery and Data Mining. AAAI Press. Menlo Park Calif. 1997. 67-73

[7] Lent B. Swami A. Widom J. Clustering association rules[C]. In IEEE 13th International Conference on Data Engineering. IEEE Computer Society Press. Los Alamitos. California. 1997. 220-231

[8] Wojcech Z. Variable precision rough set model[J]. Computer and System Science. 1993. 46(1): 39-59

[9] 刘 真, 黄兆华, 姚立文. ROUGH 集理论:现状与前景[J]. 计算机科学, 1997, 24(1): 1-5

[10] 刘胜军, 杨学兵, 蔡庆生. 关系数据库中概念层次自动提取算法研究[J]. 计算机应用研究, 1999, 16(12): 15-17

[11] Hao J W. Cai Y D. Cercone N. Data-driven discovery of quantitative rule in relation databases[J]. IEEE Transactions on Knowledge and Data Engineering. 1993. 5(1): 29-40

[12] Hu X H. Cercone N. Mining knowledge rules from databases: a rough set approach[C]. In Su S Y W ed. IEEE 12th International Conference on Data Engineering. IEEE Computer Society Press. Los Alamitos, California. 1996. 96-105

# Research on methods of mining quantitative association rules in database

*CHENG Yan, LU Tao, HUANG Ti-Yun*

School of Management, Harbin Institute of Technology. Harbin 150001, China

**Abstract**: Data mining is an important method of building intelligent decision support system. association rule is an important content of data mining. Apriori, the traditional algorithm, can only discovery the qualitative associated relation among data, but the quantitative associated relation is more helpful in decision-making. Mapping attribute's value into discrete characters is a key step in mining quantitative association rules, in which the partition granularity of attribute's value is a key factor affecting the quality of the result of data mining. In this paper, by integrating the theory of rough sets, a method of mapping attribute's value into discrete character with a fine value partition granularity is developed, and then a new algorithm of mining quantitative association rules, Apriori_2, is presented. Many rules which are helpful in decision-making can be mined by Apriori_2.

**Key words**: data mining; IDSS; association rule; rough set